Michael Esposito

Design Document 1

Project Summary

People with disabilities need a device to more easily transport their possessions. The goal of this project is to create an ATV which can independently carry cargo, reducing bodily strain on the user. The ATV will be able to navigate through varied terrain and follow a user using an onboard Intel RealSense 3D camera, capable of providing RGB and depth imagery of the surroundings in front of the vehicle.

While the original intention of the ATV was to help disabled individuals carry heavy cargo without strain on the arms, legs and back, other use cases for this project exist. Elderly individuals and individuals with temporary handicaps would benefit greatly from this technology. Similar systems are in development to increase productivity at retail stores, malls, and other marketplaces. An automated system that carries both heavy items and a large quantity of items around a store would benefit both customers and employees. This project expands upon extensive research done in the field of computer vision, user-tracking and robotics. Utilizing previous research and new technologies, the ATV will create a foundation for this new area of study.

The ATV will be able to achieve these goals of carrying heavy loads and autonomous operation using emerging technologies like the Intel RealSense Robotic Development Kit. The ATV will be able to follow a user through various terrain, over obstacles like curbs and stairs, and keep its cargo area safe and flat. An integrated computer will perform all image processing and navigation functions, allowing the ATV to follow its designated target. The three main modules involved in the ATV's functioning are user-tracking, object-avoidance, and the associated integration with hardware components.

User Tracking

Requirements

Must distinguish user	- Identify and track original user in all situations including: 1) Obstacles in between user and ATV 2) User exits camera's field of view 3) Multiple users enter field of view
Must function while robot is in motion	- Function on a moving vehicle while tracking a moving target
Must be computationally efficient	- Tracking algorithm able to run quickly so as to not inconvenience the user

A small (3.5"x2.5"x1") single-board computer capable of powering our camera and performing motor operations will be housed inside the ATV's chassis. Like the RealSense, this single-board computer (known as the UP Board) is a newly developed product, having recently entered the market after its crowdfunded origins. The camera will be housed in the front, and will have a permanent, wired connection to the UP Board. Using the Intel RealSense SDK, this computer will process the RGB and depth streams This SDK has basic user detection capabilities already, which will be modified and incorporated into the multi-threaded application initiated by ATV. The SDK has multiple language choices, but the most robust is C++. In light of this, the ATV design team has begun implementing system functions in C++. SDK modules, classes and objects will be utilized when helpful, and integrated with independently developed code from the ATV development team.

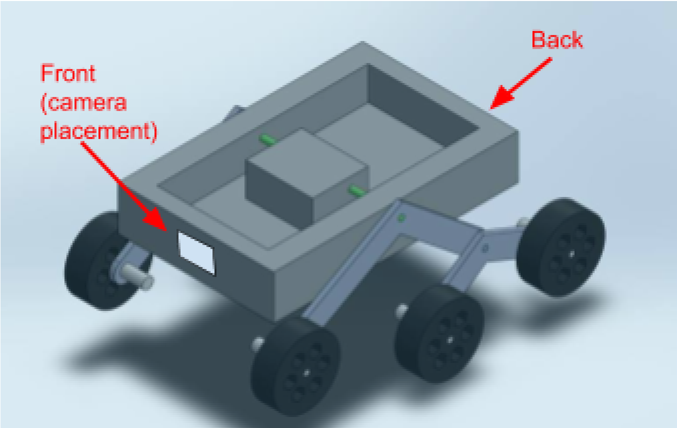

Figure 1: Rocker-Bogie Chassis

Multiple scenarios exist that make this user following system a technical innovation and achievement. The most basic use case exists when the target user walks alone and in a straight line: the ATV can constantly see the user, and maintain a specified distance away from the target using depth imagery. The user stays within the field-of-view of both the RGB and depth cameras, so no extra calculations must be performed in order to identify the location of a target user. Samples of these RGB and depth imagery streams can be seen below. This is the simplest use case, and will be the first one tested.

Unfortunately, this simplistic scenario is highly unlikely to occur during normal usage of the ATV and other use cases must be anticipated. While autonomous navigation is occurring, there may be some instances during which the ATV is unable to locate a target user. This failure may be due to environmental conditions like lighting changes, encountered obstacles, or the target user making quick changes in direction. The user may be temporarily washed out by the sun, and the ATV will need to be able to maintain course and wait for conditions to stabilize. The target user may quickly turn a corner and disappear behind a wall. Another person may walk in between the ATV and its target user, and temporarily interrupt the constantly running user tracking algorithm. In both cases, the ATV will have to rely on a stored cache of previous user positions. This cache provides the ATV with a basic understanding of object permanence, so it will recognize that the user still exists even when the user is not directly visible by either camera. This cache will be accessed when the user is not visible, guiding the ATV to the

Figure 2: RGB and Depth Streams

last known location of the user and reinitiating the user tracking process. Once the ATV reaches this location, the user-tracking thread will resume control, identify the user, and instruct the motor processes to catch up to the target user. If the target user is still unable to be found, the robot must cease movement, alert the user, and then wait until it finds the target again.

Object Avoidance

Requirements

Must be able to categorize obstacles	- Distinguish between objects that are climbable versus ones that require a route modification
Must be able to reroute ATV around obstacle	- Provides a new route around obstacle, then resumes user-tracking
Must be able to cooperate with user-tracking	- Object avoidance will override user-tracking when applicable

The difficulty in object avoidance lies within the methodologies used to identify, categorize, and define a path around obstacles. The depth camera will be the primary source of input for this operation, but input from the RGB stream may be utilized if determined to be helpful. The ATV will be able to visualize approaching objects and determine if it can climb over them, or must establish a route around them. For example, when the ATV approaches a curb, it should be able to recognize and climb over the curb. On the other hand, if the target user turns a corner behind a wall, the ATV needs to recognize its inability to pass through the wall and instead navigate around this corner. As mentioned earlier operation, the ATV will create a cache of previous known user locations. When the position of the user is unknown (such as when the user disappears behind an obstacle), the ATV must rely upon this cache to navigate to a point where it can resume user-tracking. The user tracking and object avoidance processes will need to be threaded and concurrently running. An example of this occurs when the ATV is avoiding an obstacle: the ATV is relying on previously gathered data, but how will it know when the user reenters the field-of-view? User tracking will be running at all times, and the two processes will share set of variables or mutex locks to determine which process has control over the critical section of motor control code. A preliminary diagram of this interaction can be seen in Figure 3.

Figure 3: Program Flow

Integration with Other ATV Components

The user tracking and object detection modules are important, but their entire purpose is to provide direction to the motor control system. Therefore, the interaction of these three modules is of great importance to the overall function of the ATV. At the highest level, this software and hardware interaction is illustrated in the following diagram. The primary method of obtaining input data is from the Intel RealSense R200. This camera will capture RGB data while simultaneously capturing images from two stereoscopic infrared

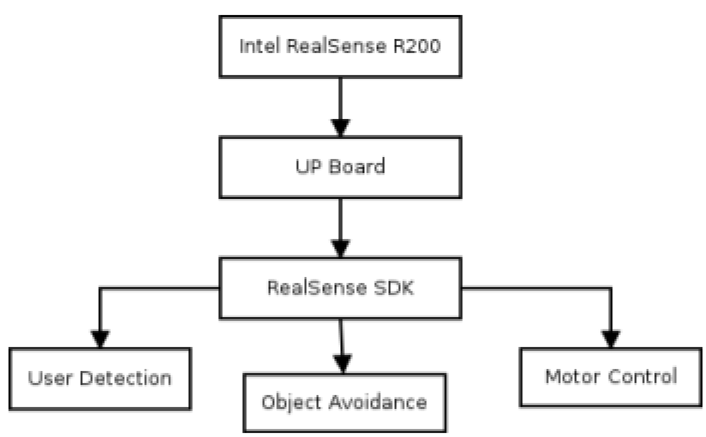

Figure 4: Hardware/Software Interaction

cameras. It then uses these two images to produce the depth image, which can then be used to detect users and objects.

User Interface

The ATV will have a limited user interface. Ease and simplicity of use were two important design specifications. Therefore, the ATV will be outfitted with the bare minimum—a switch to open/close the cargo compartment and an on/off switch. The target user will need to first open the cargo compartment, place his/her cargo inside/ and then close the compartment. Next, he/she must turn on the ATV to begin user tracking and object avoidance after a brief initialization period.

Project Timeline

In order to meet required deadlines, the ATV team has created and will adhere to the following research and development timeline. The selected dates will enable the development team ample time to answer questions identified during preparatory phases. Future work is pending a new investment, but keeping aligned with these proposed dates should not pose an issue.

	Task Name	Duration	Start Date	End Date	Predecessors	% Complete
1	Project Initiation	24d	09/09/16	10/12/16		100%
2	Camera Configuration	49d	10/04/16	12/09/16		80%
3	User Tracking	30d	11/14/16	12/23/16		40%
4	Object Avoidance Algorithm Implementation	26d	01/17/16	02/19/16		0%
5	Chassis/Compartment Construction	25d	01/17/17	02/20/17		0%
6	Component Integration	16d	02/19/17	03/10/17		0%
7	Testing and Troubleshooting	24d	03/13/17	04/13/17		0%

Figure 5: R&D Plan and Timeline

Constructing and designing the ATV will require a lot of experimentation in order to refine the underlying algorithms. In order to ensure that the the ATV fulfills all requirements, specifications, and use cases, the ATV will be tested in the various locations and environment types. The ATV must be able to operate in urban, suburban and rural settings to maximize profitability, and copious amounts of test data will need to be gathered to ensure desired functionality in a diverse set of environments. Through a series of iterative testing practices and demonstrations, the progress of migrating the ATV from planning to production phases can be accomplished easily. By the first quarter of 2017, the ATV team hopes to have a fully-functional prototype.